US-PAT-NO: 6446137

DOCUMENT-IDENTIFIER: US 6446137 B1

TITLE: Remote procedure call system and method for RPC mechanism independent client and server interfaces interoperable with any of a plurality of remote procedure call backends

---------- KWIC ---------

Detailed Description Text - DETX (60):
For each remote function defined in the IDL. a client stub routine is generated. Calling this stub function causes the appropriate server function to be invoked with the same arguments: the result of the server routine is returned by the client stub routine. int opsp.fwdarw.method (appl_optab_t*opsp, args . . .) Function Make an RPC call. Invokes function method as defined in the IDL. opsp is a pointer to an appl_optab_t as initialized by vrpc_begin_buf( ) (see Chapter 1 [Client API]. page 1). args are the actual RPC arguments. Note that the VRPC client stub routines all accept arguments as pass-by-reference arguments. All non array arguments are passed by reference. Arrays are also passed by reference. but no additional translation is performed since C already passes arrays by reference. Also note that return function values are passed back in the last argument of the argument list. In other words. the function value is converted to a pass-by-reference variable. method( ) returns 0 on success and -1 on error.
Each `appl_vclnt.c` file contains an interface definition.

The definition is
of type appl_optab_t and is conventionally placed in a
variable called
APPL_optab. appl_optab_t Data type typedef struct [
optab_base_t base; int
(*func) (appl_optab_t*, const func_arg*, int*func_res); .
. .] appl_optab_t;
base is of type optab_base_t as described below (see
Section 3.2 [VDR
Internals]. page 4). func each appl_optab_t has one or
more members which are
function pointers. These are initialized by the compiler
generated code to
client stubs which perform the corresponding RPC. The
appl_optab_t has
function pointers to the client stubs. Each of these
client stubs (ails the
be_send_call routine for that appl_optab_t with the
appropriate arguments. One
of these arguments is a vrpc_proc_spec_t. This **page under construction**.
Chapter 2: VRPC Server Interface 3

Detailed Description Text - DETX (63):
These functions are declared in the following header
file: #include
<vrpc/vrpc.h> void*vrpc_begin (const char*name, const
void Function
*clnt_ops) Establish VRPC connection. name is used to
query a naming system to
identify and locate the server process. clnt_ops
identifies the interface
being accessed. It is the address of a appl_optab_t as
found in the compiler
generated file `appl_vclnt.c`. vrpc_begin( ) returns a
pointer to an
initialized appl_optab_t. or nil on error. This **page under construction**.
Chapter 3: Interface Specification Mechanism 4

Current US Original Classification - CCOR (1):
709/330

US-PAT-NO: 6253208

DOCUMENT-IDENTIFIER: US 6253208 B1

TITLE: Information access

---------- KWIC ---------

Detailed Description Text - DETX (34):
Preferably, the database 135 may be arranged to record a minimum set of related information about each property for sale or for rent, likely to be common to all commercial property extracted from the temporary results cache 150 by the result analyser 130. The scope of information stored may be sufficient to support no more than a basic commercial property search of the database 135. Preferably, a publicly accessible Internet interface may be provided to the information access system database 135 including, in a commercial property trading application, means for a user to enter a query defining basic property characteristics. Such a query interface is provided, for example, in the Applicant's "PropNet" property trading service for the Internet, published in the applicant's "BT Technology Journal", Volume 15, No. 2, April 1997, a public trial system being made available on the Internet at http://transend.labs.bt.com/BTPropNet. A PropNet user may submit to the PropNet query interface a simple profile of the type of property being sought, specifying only property type, location and floor area for example. The query interface, in turn, uses the submitted profile to search the information access system database 135 for matching property and presents the results to the user as a summarised shortlist of properties. Where an

associated HTTP **URL** is
recorded in the **Real Estate** file 300, the query interface
may display an **HTML**
"hot-spot" to enable a user to "hyperlink" to the third
party web page
identified by that **URL** and to view full details on the
property, including any
other information on the property such as still or
interactive video images
made available by the advertiser.

Current US Cross Reference Classification - CCXR (3):
705/26

US-PAT-NO: 6546400

DOCUMENT-IDENTIFIER: US 6546400 B1

TITLE: Method and system for creating trading cards

---------- KWIC ---------

Detailed Description Text - DETX (46):
A list of available templates may be presented to the user through a web site. The user may then select desired templates for the trading card. Statistical data and images transferred by the user to the central server 1102 may be imported in the template for creation of the trading card. The user may then perform any desired final **editing** to the trading card via the **web site** The finalized card may be stored in a card file in a card database or data file accessible by the central server 1102. The user may e-mail a copy of the card file to a user device 1104. Of course, the user may also download and save a copy of the card file in the local memory 1116 of the user device 1104. On the local user device, the user may designate the card file as "wallpaper."

Current US Cross Reference Classification - CCXR (2):
707/9